



Timing Interruptions for Better Human-Computer Coordinated Planning

Citation

Sarne, David and Barbara Grosz. 2006. Timing interruptions for better human-computer coordinated planning. Paper presented at 2006 AAAI Spring Symposium on Distributed Plan and Schedule Management in Palo Alto, CA, March 27–29, 2006.

Published Version

<http://www.aaai.org/Papers/Symposia/Spring/2006/SS-06-04/SS06-04-026.pdf>

Permanent link

<http://nrs.harvard.edu/urn-3:HUL.InstRepos:2562373>

Terms of Use

This article was downloaded from Harvard University's DASH repository, and is made available under the terms and conditions applicable to Other Posted Material, as set forth at <http://nrs.harvard.edu/urn-3:HUL.InstRepos:dash.current.terms-of-use#LAA>

Share Your Story

The Harvard community has made this article openly available.
Please share how this access benefits you. [Submit a story](#).

[Accessibility](#)

Timing Interruptions for Better Human-Computer Coordinated Planning

David Sarne, Barbara J. Grosz

Division of Engineering and Applied Sciences
Harvard University, Cambridge MA 02138 USA

Introduction

Recent developments in the area of automated systems for planning and scheduling the activities of both individuals and coordinated groups demonstrate the potential for autonomous agents to collaborate with and support teams of people operating in highly dynamic environments (Wagner *et al.* 2004; Schurr *et al.* 2005; Scerri *et al.* 2003). A wide variety of mission-critical domains exhibit such highly dynamic environments, ranging from first responders (Wagner *et al.* 2004; Schurr *et al.* 2005), through exploration of remote planets and cleanup of hazardous sites to terror theaters and military conflicts (McClure 2000). In such environments, autonomous agents (which we will refer to as *coordinators*¹), that are designed to efficiently process the rich set of information affecting the planning problem (Myers *et al.* 2002), have the potential of suggesting alternative courses of action (given limited time for decision making and changing plans) which more effectively achieve mission objectives. Such coordinators can take on responsibility for changes in task schedules, thus freeing people involved in doing tasks from being unnecessarily overburdened, especially in times of crisis when plans need to be adapted to new circumstances (Schurr *et al.* 2005; McClure 2000).

While intuitive human-computer interfaces and intelligent team coordination are important, critical failures will occur unless human-agent interaction is appropriately managed (Schurr *et al.* 2005). A fundamental requirement for any system that is to collaborate with people is to have the appropriate division of labor (Shieber 1996); the coordinator must know which decisions it can make and when to defer to or consult people on the team (or other agents). Furthermore, to be effective, coordinators must be able to adapt appropriately to the skills and knowledge of their human teammates; that is, they must have adjustable autonomy (Scerri *et al.* 2003).

Given the increased complexity and dynamic stochastic nature of events and changes in such domains (as well as the potential for schedule failures to be costly or catastrophic), many proposed implementations of coordination and planning systems rely on a distributed design, emphasizing scalability and responsiveness (Wagner *et al.* 2004). In these settings, individual coordinators are associated with specific participants (team leaders, units commanders, which we will refer to as *owners*) in the organizational hierarchy.

The design of coordinators must take into consideration several important factors. First, the goal of coordinators is to maximize an overall team-objective and pre-defined utility rather than to maximize the specific preferences or utility of an in-

dividual owner. Consequently, the focus of interactions is on facilitating a collaborative effort towards reaching the defined objectives. Second, the people involved in these tasks and the coordinators themselves typically have limited resources. Since each interaction between a coordinator and its owner incurs costs for both parties, the process of initiating and managing an interaction with the user needs to take into account the cost-effectiveness of the interaction. Costs include both the potential degradation in the task the person is currently executing caused by the interruption and the loss of utility that could be achieved if the computational resources deployed for the interaction were used instead by other system components. On the utility side, there is the potential increase in (actual) performance obtained as a result of the interaction. Thus, coordinators must be able to manage efficiently partial, perhaps non-optimal, but shorter and less resource consuming interactions with the user.

We argue that a coordinator's performance can be significantly improved by making use of three principle sources of value added by the abilities of the user to (1) reduce the search space (e.g. by eliminating paths leading nowhere); (2) rank alternatives produced by the system, estimating the probability of successfully implementing them in the environment; and (3) supply additional information thus relaxing constraints imposed by the system's limited knowledge of the environment, when generating alternative plans. Each of these added-values draws on knowledge the person has that is beyond that available to the system. In this context we present a high-level overview of our design of a Coordination Autonomy (CA) module as part of the cMatrix project². Within this framework, the CA module bears the responsibility to intelligently initiate and manage the necessary interactions with the owner.

Designing an Efficient Intelligent CA Module

Common to all three sources of user added-value is the fact that users' experiences play a significant role in their ability to be of help to the system. Using their experience people can successfully analyze scenarios that they have not previously directly faced, deduce and adapt appropriate solutions to slightly different problems and anticipate the results of actions taken. Furthermore, an owner may possess information that the system may not have. Though part of this information is based on accumulated experience (e.g. understanding the influence of weather on the probability of success of executing an action), most of this information is gathered from parallel (formal and informal) communication channels which people naturally scan. Typical sources of such information are

¹We adopt the name from the DARPA COORDINATORS program, aimed to design, implement, and demonstrate a computational framework for distributed planning.

²The cMatrix (for Coordinated Multi-Agent Team Reasoning and Incremental Execution) is an R&D project under the DARPA COORDINATORS program, built to support response to events that affect the ongoing military plan.

occasional coordination meetings (e.g., used for reporting status of task execution), open communication the user passively listens to (e.g. when leaving the radio open, one gets to listen to messages associated with other teams in the area) and direct communication used for coordination with other people acting on a joint task (throughout which individual often informally learns about the status and existence of other actions being executed by others). As automated coordinators and planners evolve, their ability to scan various communication channels will significantly improve, and thus the added-value from user interactions will derive mostly from accumulated experience.

Recognizing the particular sources of user added-value, and given environmental properties in which coordinators operate, the design of a successful coordination autonomy system should emphasize such functionalities as (1) presentation of plans (e.g., focusing on elements that are relevant to that user at that time); (2) interruptibility (e.g., knowing when it is useful and appropriate to interrupt a user); and (3) user involvement (e.g., support effective introduction of the problematic elements of a plan, allowing the user to add new plans and to modify them quickly). The above functionalities enable the system to take advantage of human strengths in a responsive process which evaluates utilities and costs from a global point of view.

Our proposed CA module has two main interfaces, shown in figure 1. The first connects to the other cMatrix modules, which are used both as a source for learning about the changing environment and for evaluating and analyzing the quality of different plans of action. The second interface is with the user (the UI layer), supporting all the required different kinds of presentation methods and user-machine interaction functionalities. This layer is based on basic and established UI planning-related information representation concepts (e.g. clustering of related activities, drill-down capabilities, navigation within a plan). The information to be presented using these functionalities, its extent and detail level, the order by which it is presented and the transition between the different representation modes, are all influenced by other, hidden elements of the module. These components include the *Environment Modulation* (responsible for maintaining a continuous representation of the environment as known to the cMatrix and the resulting complexities in the current plan accordingly), the *User Modulation* (responsible for maintaining a continuous representation of the environment, including monitoring the user's state in order to assess the costs of interrupting him), the *User Profiling* (responsible to monitoring the interactions with the specific user and refine the interaction methods used according to an accumulated repository of known behaviors), *learning algorithms* that guarantee improvement over time and, finally, an *Interaction Manager* which filters and integrates the inputs received from all the other objects into an optimal interruption strategy, defining the method, order and structure of the

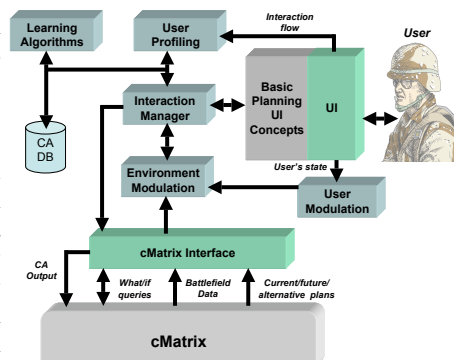


Figure 1: Data-flow diagram

interaction with the user.

interaction with the user.

As part of the process of assessing gain in interrupting the user, the system takes into consideration the information the user already has about the planning problem, the complexity of the problem, the user's capabilities and the extent of the planned interaction. Once the *Interaction Manager* evaluates the gain to be greater than the expected cost of the interruption, it initiates an interruption. The interaction process has three principle subsequent stages as shown in figure 2. Estimating the gap between the user's familiarity with the planning problem and its actually extent, the CA module efficiently navigates the user throughout the relevant aspects of the problem, focusing the user's attention on the newest and most relevant information received. Then, the CA module supports a cooperative planning process through a combined sequence of different interaction mechanisms, until it finally feeds back to the cMatrix planning elements the user input.

The CA module integrates learning algorithms for demonstrating improvement over time. The learning process is two-fold. First, it captures specific user characteristics (emphasizing issues such as strengths and weaknesses, preferences, capabilities and specific behaviors). Second, it supports the process of modeling the environment the user is operating in, in particular the way the user perceives it.

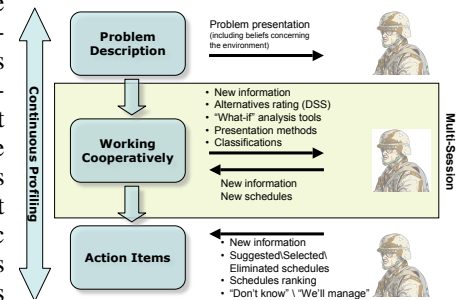


Figure 2: CA-User interaction flow

Conclusions

The high operations tempo and growing complexity of planning (and re-planning) in various mission-critical domains suggest an approach in which systems act as primary planners rather than assisting the user in planning. A critical factor in such systems success is their capability to initiate, maintain and manage intervention sessions with the commanding user. User involvement may offer great benefit, but it is a costly process that needs to be managed wisely. The suggested design of the CA module, as part of the cMatrix system, serves as an innovative framework for managing such user involvement.

References

- McClure, W. 2000. Technology and command: Implications for military operations in the twenty-first century. Maxwell Air Force Base, Center for Strategy and Technology.
- Myers, K. L.; Jarvis, P.; and Lee, T. 2002. Active Coordination of Distributed Human Planners. AIPS, 63-71.
- Scerri, P.; Pynadath, D. V.; Johnson, W. L.; Rosenbloom, P. S.; Si, M.; Schurr, N.; and Tambe, M. 2003. A prototype infrastructure for distributed robot-agent-person teams. In AAMAS, 433-440.
- Schurr, N.; Marecki, J.; Lewis, J. P.; Tambe, M.; and Scerri, P. 2005. The defacto system: Training tool for incident commanders. In AAAI, 1555-1562.
- Shieber, S. M. 1996. A call for collaborative interfaces. *Computing Surveys* 28A (electronic).
- Wagner, T.; Phelps, J.; Guralnik, V.; and VanRiper, R. 2004. An application view of coordinators: Coordination managers for first responders. In AAAI, 908-915.